Statistical Process Control for Software

Anita D. Carleton

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Sponsored by the U.S. Department of Defense

Topics

Introduction

Statistical Thinking and Process Thinking

Understanding Variation

Control Charts

Software Example

10 Steps to Get Started and FAQs

Process Thinking

Focus on the processes to improve quality and productivity

Managers must focus on fixing processes, not blaming people

Management action uses data from the process to guide decisions

Recognize that variation is present in all processes and that it is an opportunity for improvement

Statistical Thinking

Fundamental axioms

- **all work is a series of interconnected processes**
- **all processes are variable**
- **understanding variation is the basis for management by fact and systematic improvement**

Source: G. Britz, "How to Teach Others to Apply Statistical Thinking," *Quality Progress*, June 1997.

Interpretation Requires Analysis

Input *Transformation* *Output*

Separation of signal from noise requires rigorous analysis pr

This allows inferences to be drawn to guide decisions and ac

Understanding Variation

"While every process displays variation, some processes display controlled variation, while others display uncontrolled variation."
- Walter Shewhart

Common cause or "controlled" variation - due to normal or inherent activities among the process components

Assignable cause or "uncontrolled" variation - due to anomalies in the process

total variation = common cause variation + assignable cause variation

Process Behavior—Variation and Stability

Variation = process noise + process anomalies

Stable process = ~~process anomalies~~

Stable process = Controlled process

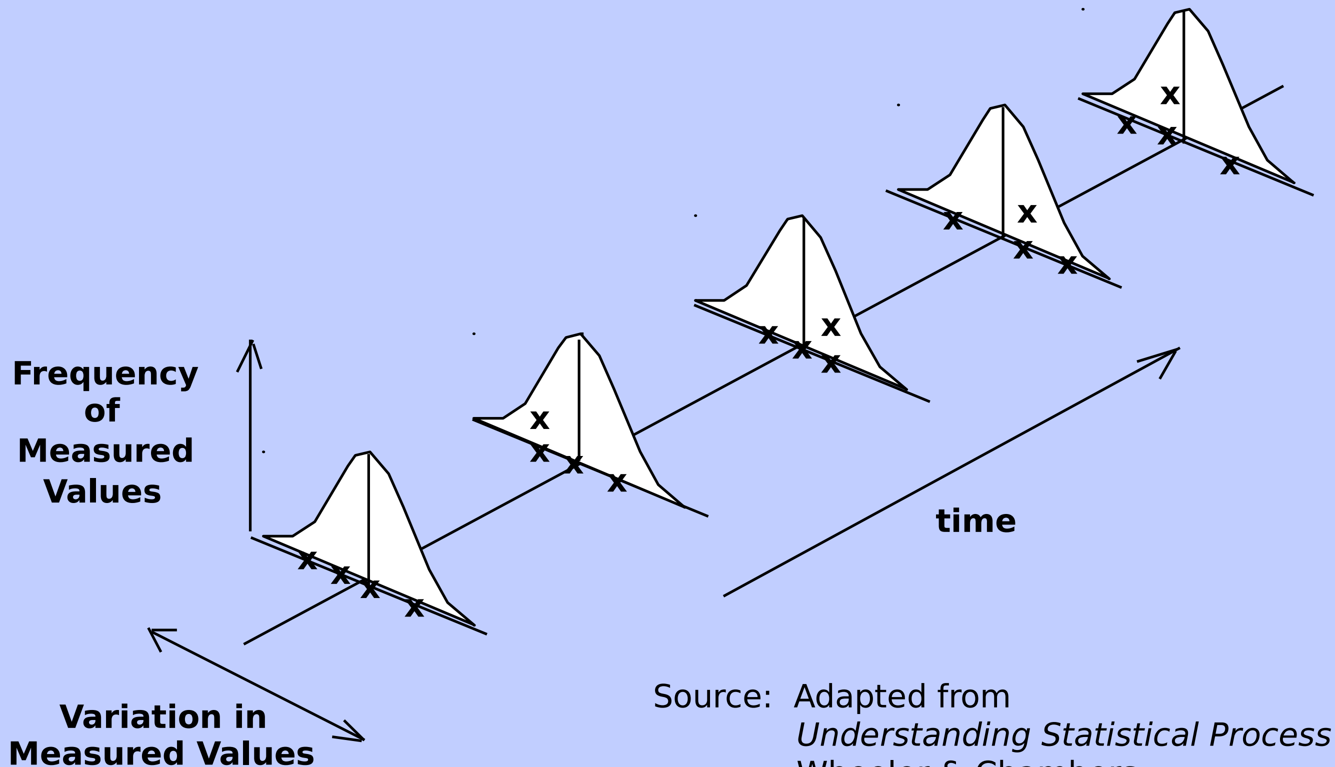
Stability

Is the process that we are managing behaving predictably?

Stable process = process in statistical control

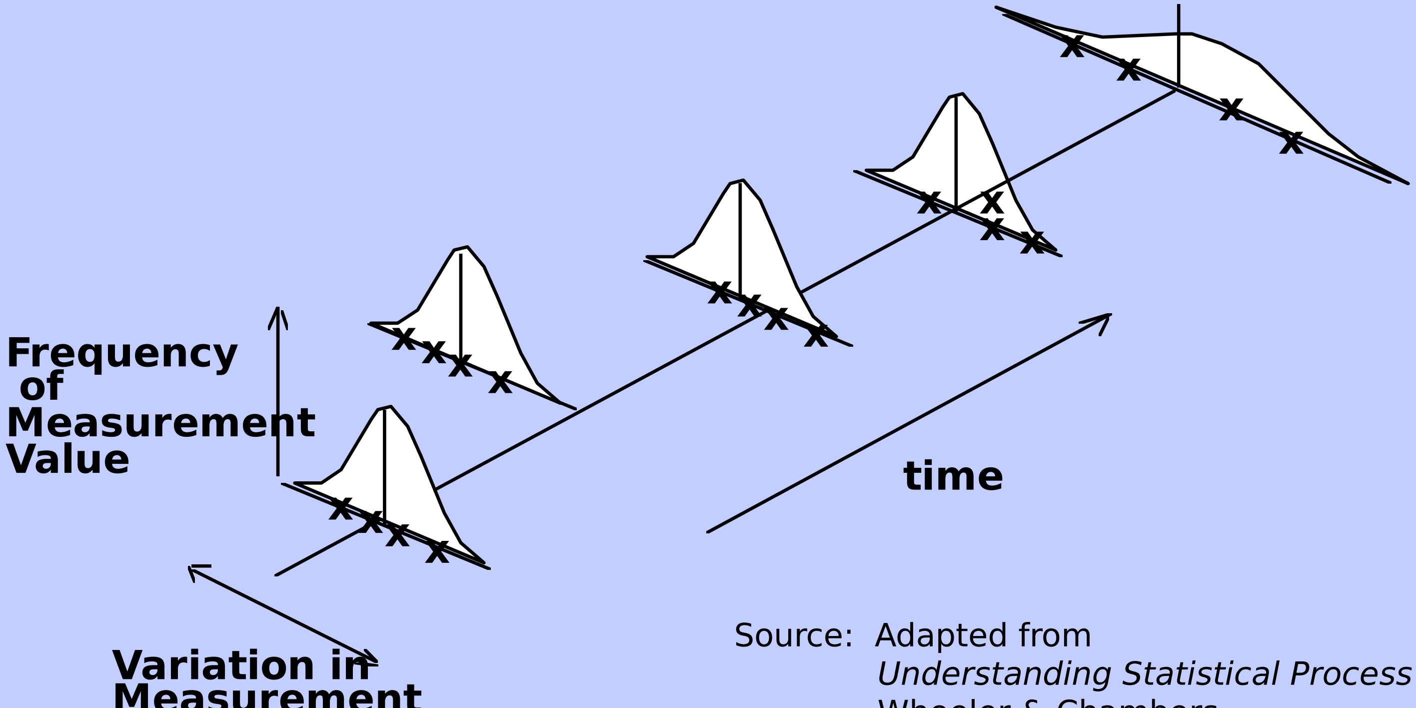
= sources of variability due to common causes

The Concept of Controlled Variation



Source: Adapted from *Understanding Statistical Process Control*, Wheeler & Chambers

The Concept of Uncontrolled Variation

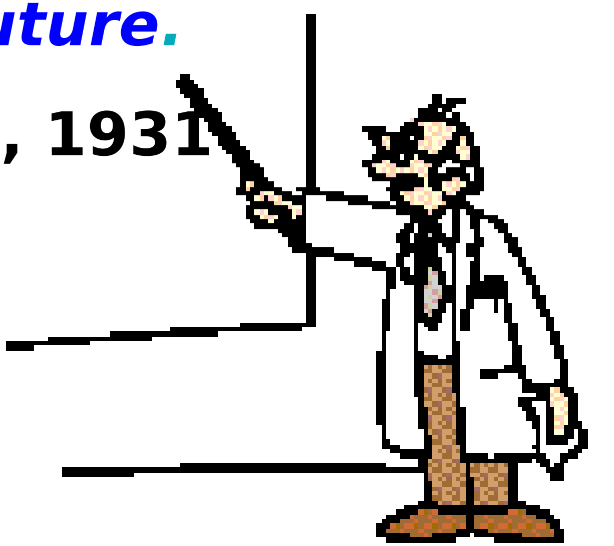


Source: Adapted from *Understanding Statistical Process Control*, Wheeler & Chambers

Statistical Process Control

A phenomenon will be said to be controlled when, through the use of past experience, we can predict, at least within limits, how the phenomenon may be expected to vary in the future.

Walter A. Shewhart, 1931



Why SPC for Software Development?

To understand the "reliability" of human processes

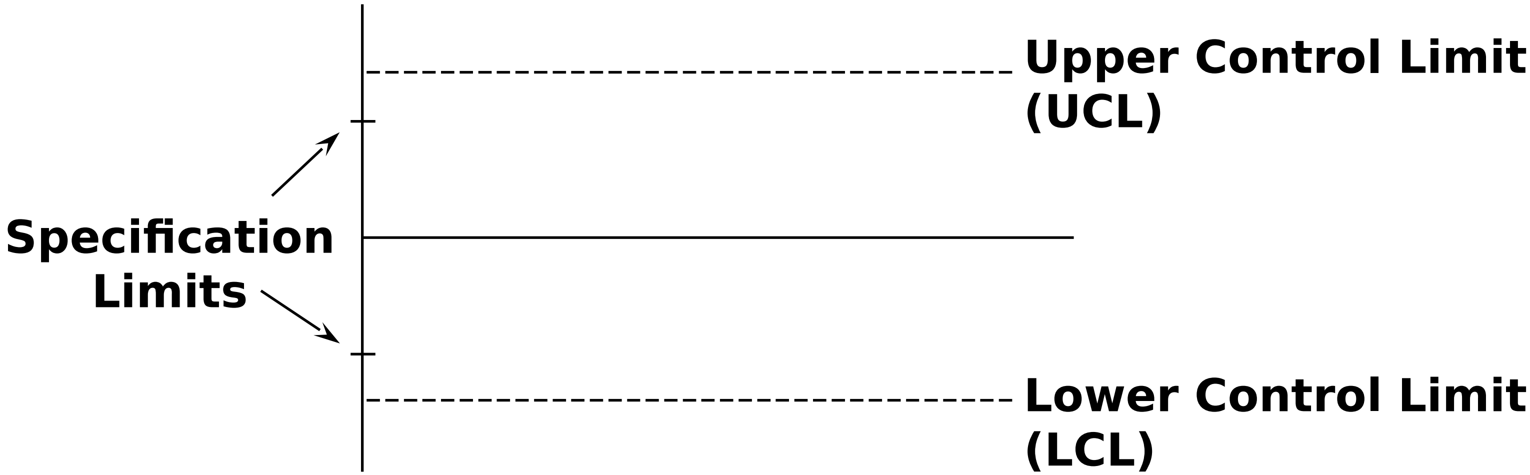
To establish bounds on management expectations

To understand patterns and causes of variations

To validate metric analysis used to forecast and plan

Source: T. Keller, "Applying SPC Techniques to Software Development--A Management Approach," 1999 SEPG Conference.

Control Charts - 1

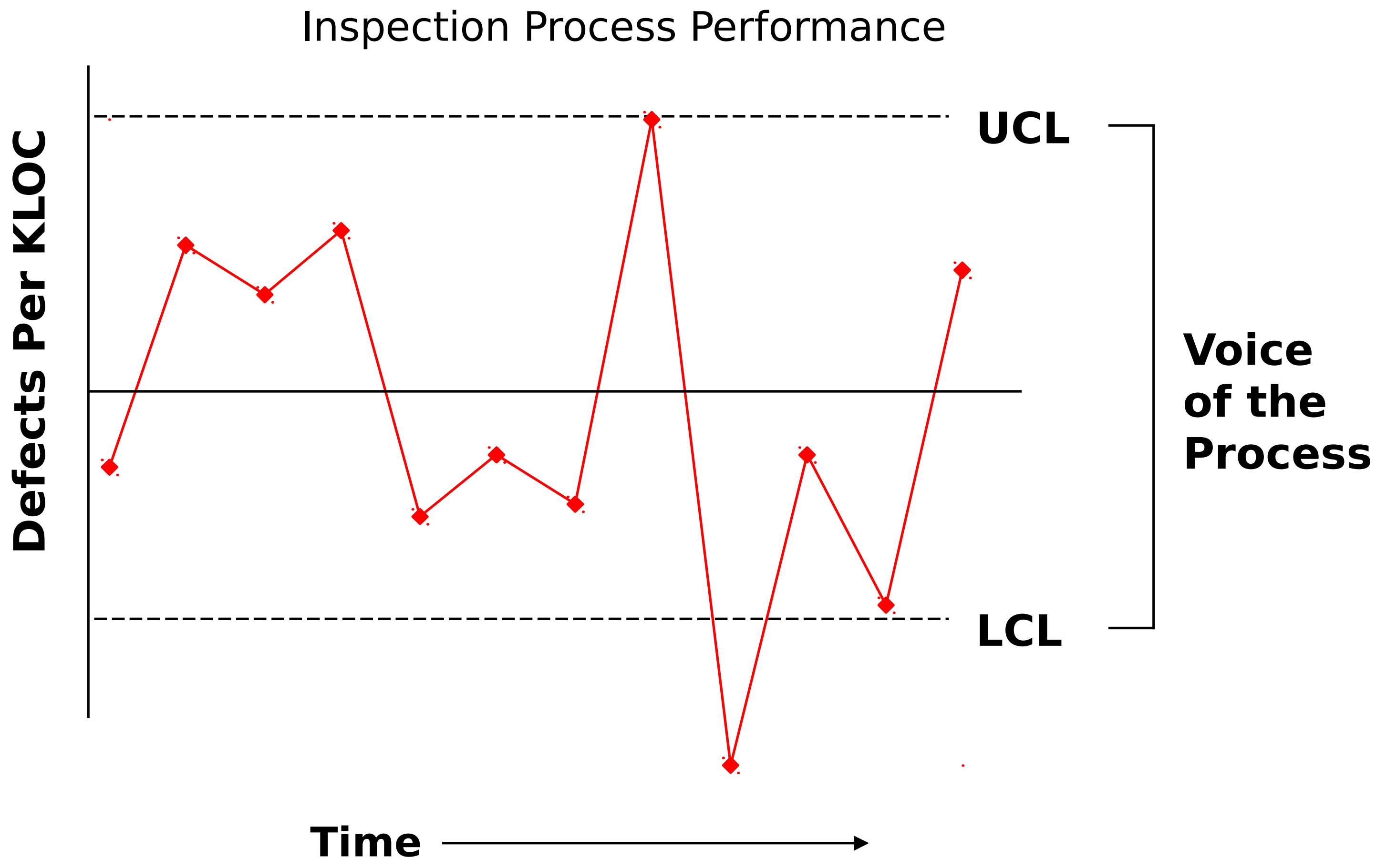


Limits

Control Limits → **Determined by Process Performance Measurer**
(Voice of the process)

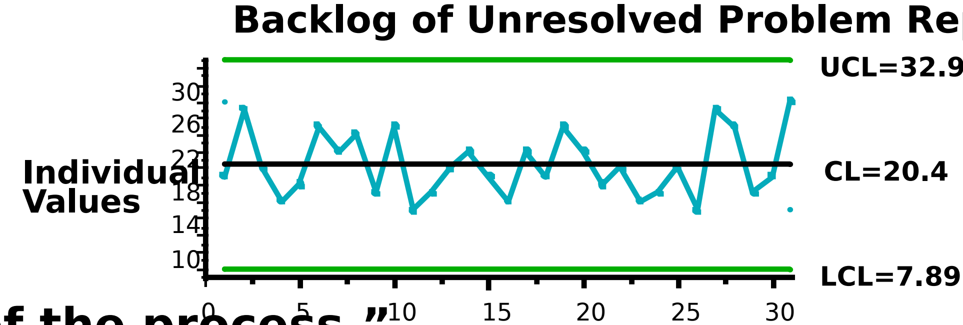
Specification Limits → **Set by customer, engineer, etc.**
(Voice of the customer)

Control Charts - 2



Why Control Charts?

Control charts let you know what your processes can do, so that you can set achievable goals.



They represent the "voice of the process."

Control charts provide the evidence of stability that justifies predicting process performance.

Control charts separate signal from noise, so that you can recognize a process change when it occurs.

Detecting Instabilities and Out-of-Control Situations - 1

To test for instabilities in processes, we examine control charts for instances and patterns that signal nonrandom behavior.

Values falling outside the control limits and unusual patterns within the running record suggest that assignable causes exist.

Detecting Instabilities and Out-of-Control Situations - 2

Test 1: A single point falls outside the 3-sigma control limits.

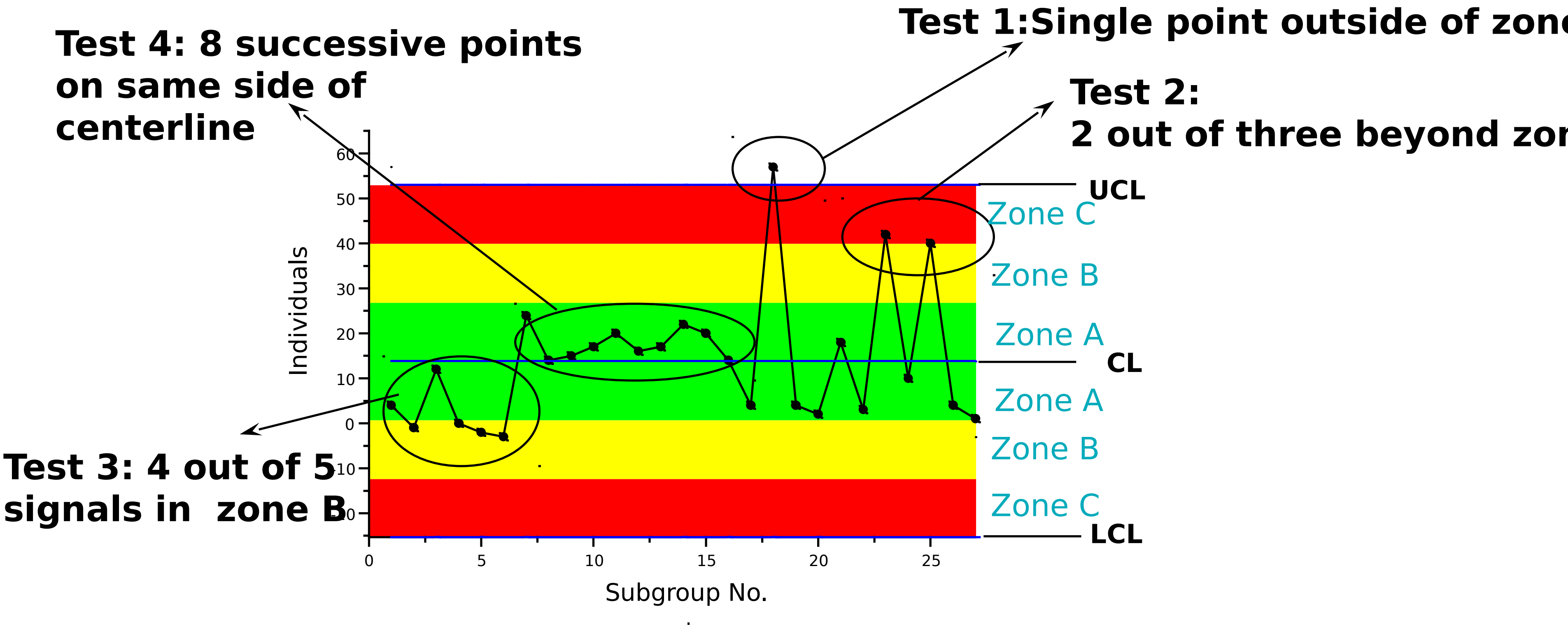
Test 2: At least two of three successive values fall on the same side of, and more than two sigma units away from, the center line.

Test 3: At least four out of five successive values fall on the same side of, and more than one sigma unit away from, the center line.

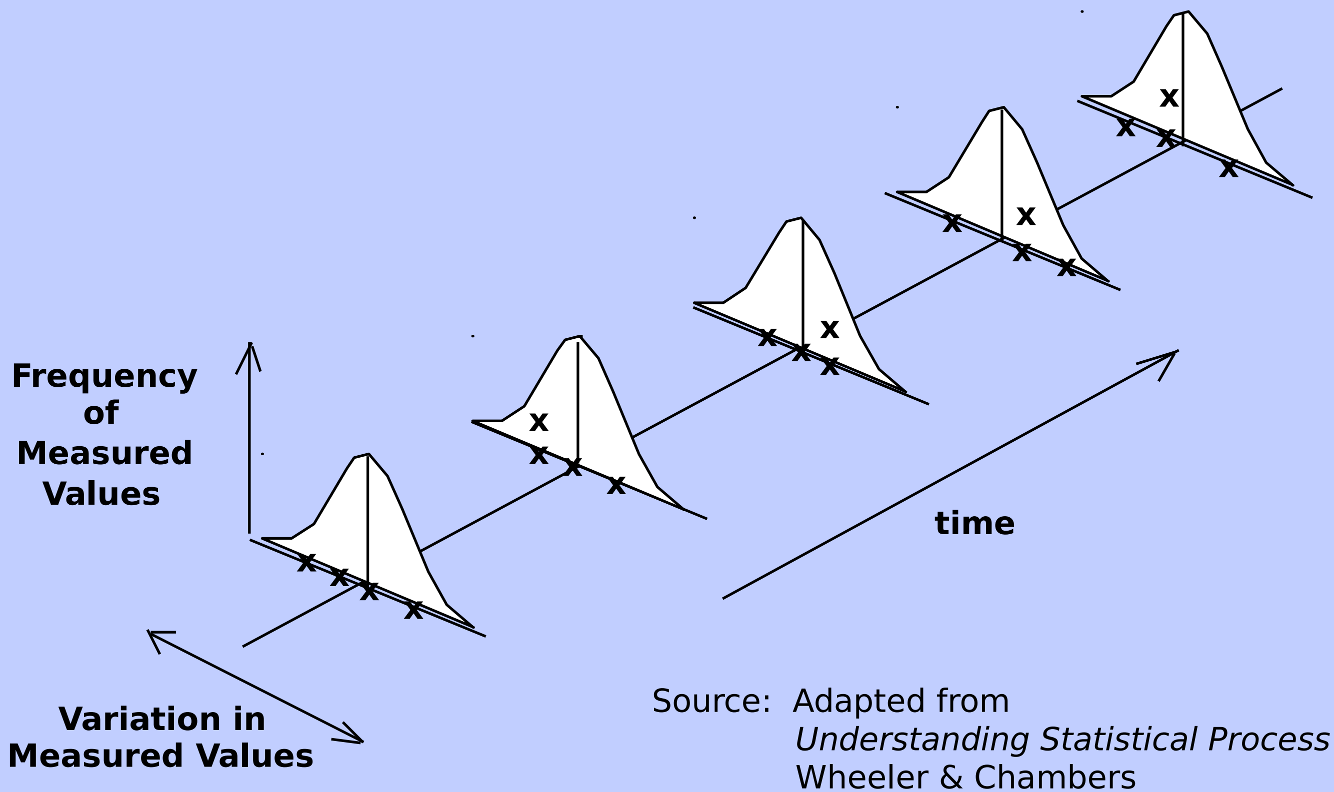
Test 4: At least eight successive values fall on the same side of the center line.

Source: Western Electric Handbook

Detecting Instabilities and Out-of-Control Situations - 3

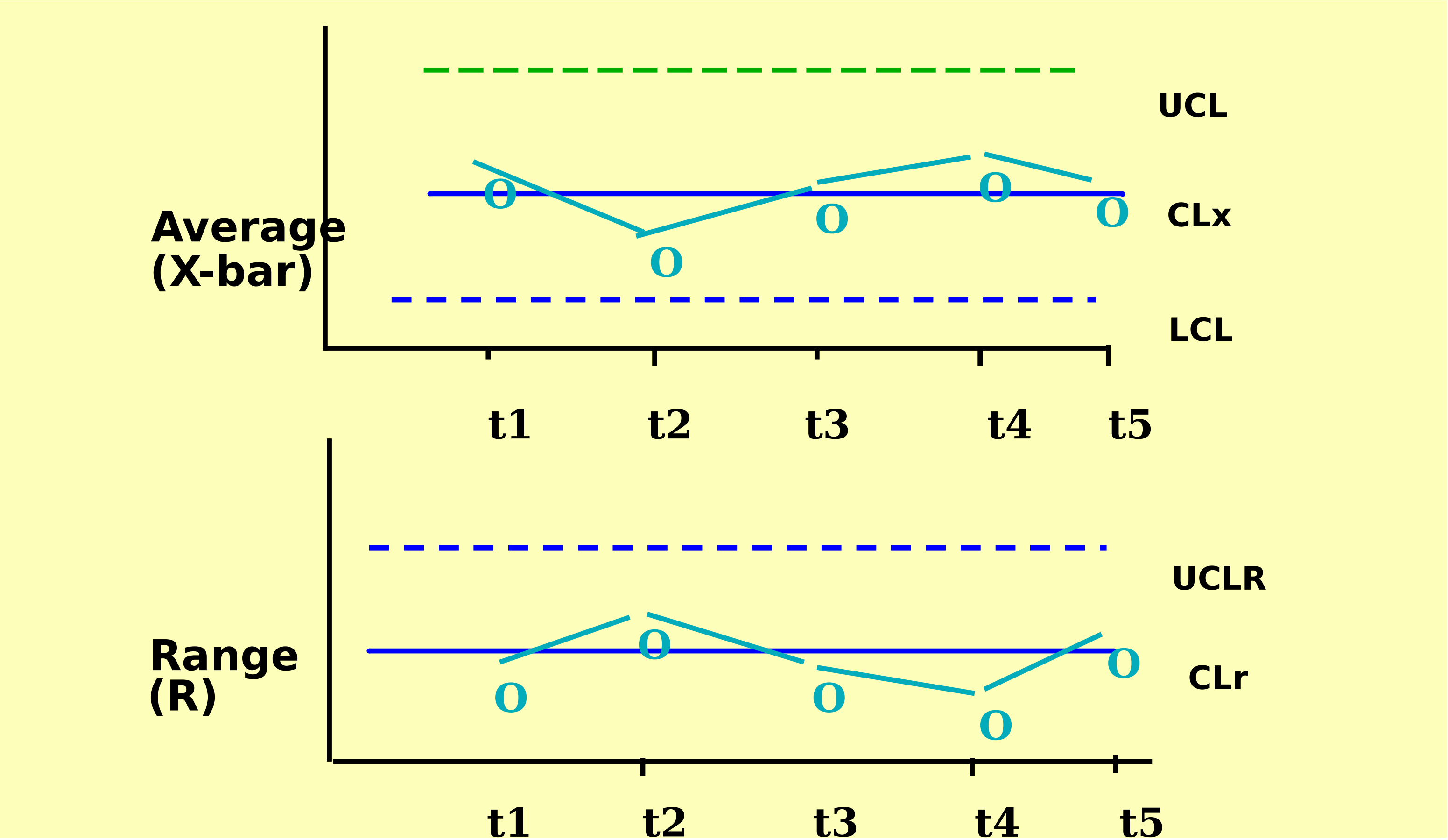


An Example Process in Statistical Control



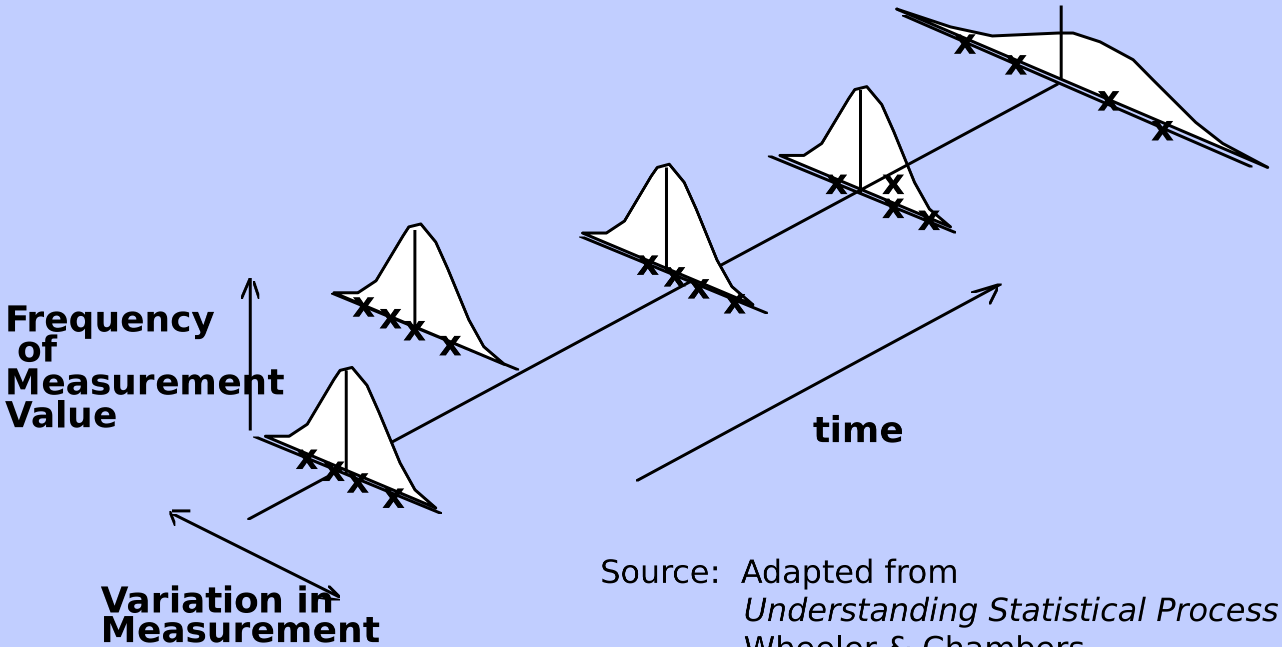
Source: Adapted from
Understanding Statistical Process Control,
Wheeler & Chambers

X-Bar and Range Charts for a Process that Is In Control



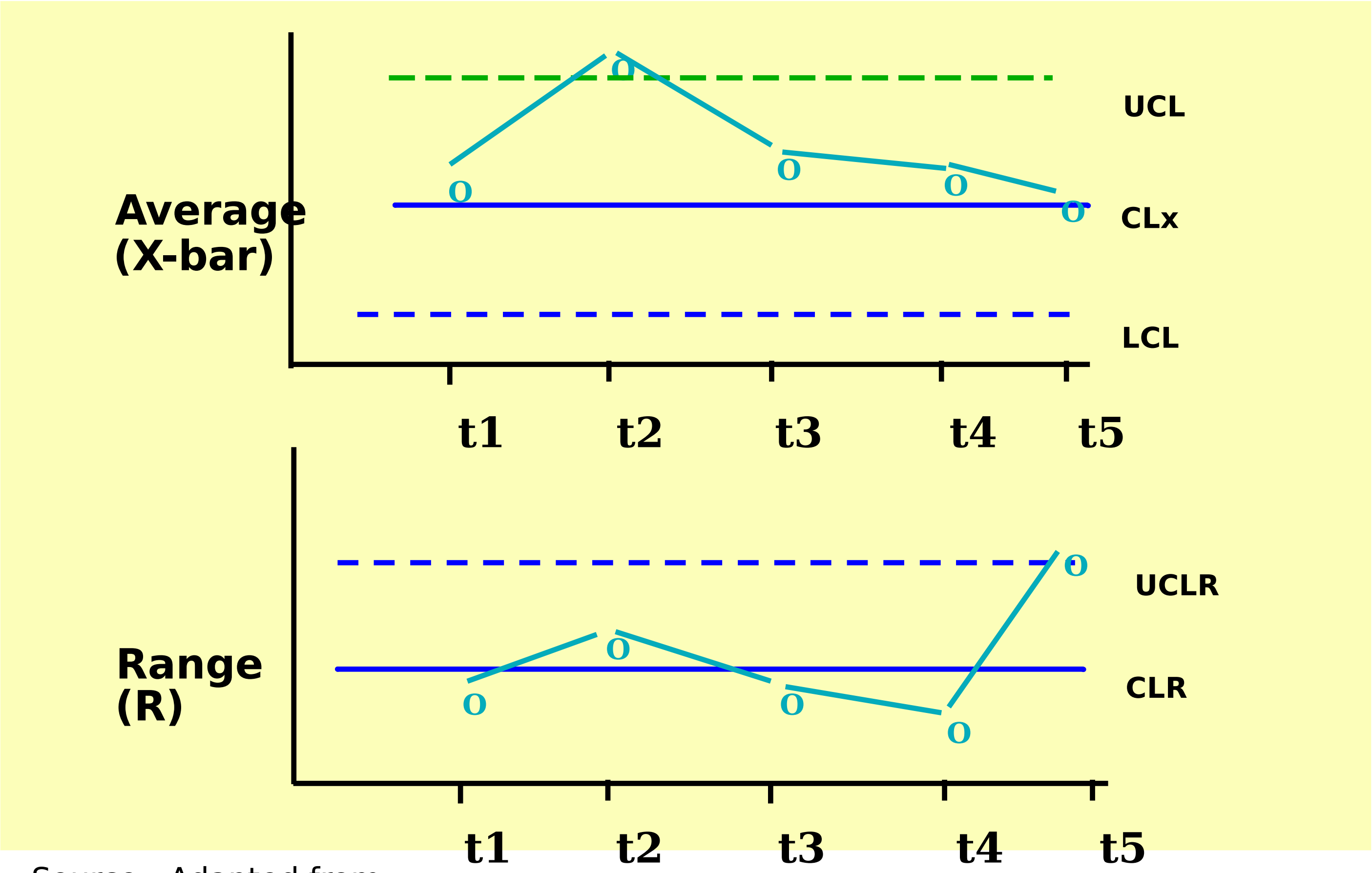
Source: Adapted from *Understanding Statistical Process Control*, Wheeler & Chambers

An Example Out-of-Control Process



Source: Adapted from
Understanding Statistical Process Control,
Wheeler & Chambers

X-Bar and Range Charts for a Process that Is Out of Control



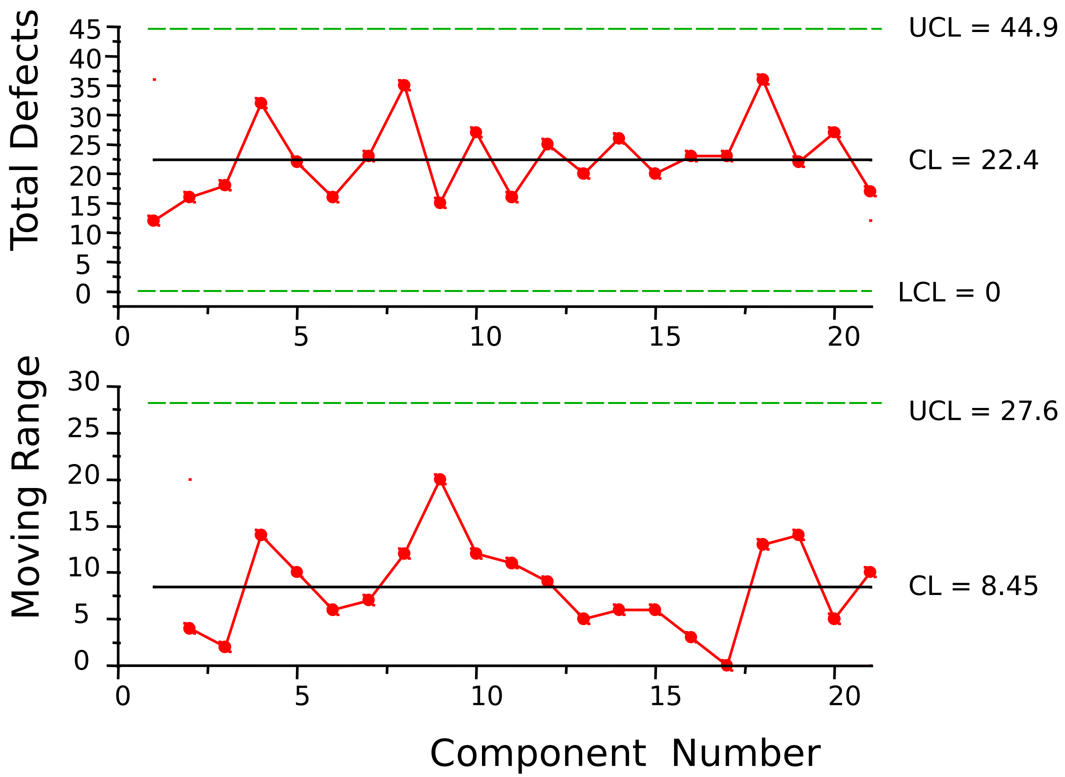
Source: Adapted from
Understanding Statistical Process Control,
Wheeler & Chambers

Example: Summary of Defect Types Found During Component Inspections

Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	**Totals**
Defects	12	16	18	32	22	16	23	35	15	27	16	25	20	26	20	23	23	36	22	27	17	471
Defect Type	Number of Defects per Type per Component																					
Function	3	5	4	4	4	3	3	20	4	11	2	3	3	5	3	7	4	5	5	15	2	115
Interface	2	2	4	4	3	4	2	3	3	4	2	3	5	3	3	3	2	16	6	2	4	80
Timing	1	1	0	1	1	0	2	1	0	0	2	0	1	1	1	1	1	0	1	0	0	15
Algorithm	0	0	1	14	2	0	0	0	0	0	0	1	5	2	7	6	5	1	2	0	1	47
Checking	1	1	5	1	7	1	1	2	0	1	6	3	1	12	1	0	2	4	3	5	2	59
Assignment	0	2	0	4	1	2	1	3	2	3	2	8	1	0	2	1	2	1	0	1	1	37
Build/Pkg.	3	1	1	2	1	0	0	4	3	6	1	0	2	1	1	1	3	2	2	2	1	37
Document	2	4	3	2	3	6	14	2	3	2	1	7	2	2	2	4	4	7	3	2	6	81

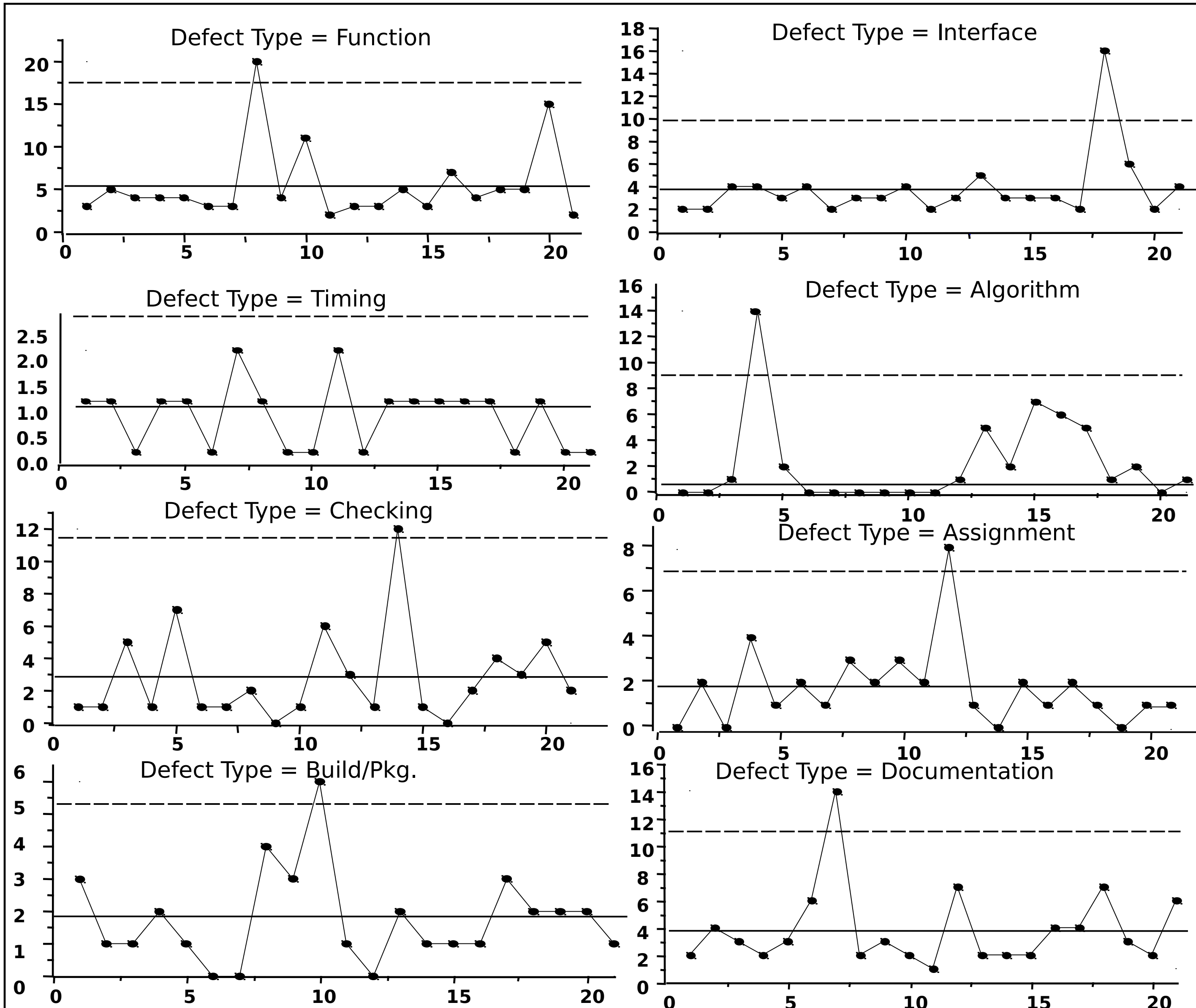
Source: W. Florac and A. Carleton, *Measuring the Software: Statistical Process Control for Software Process Imp*
June 1999.

XmR Charts for the Total Number of Defects Found in Component Inspectio



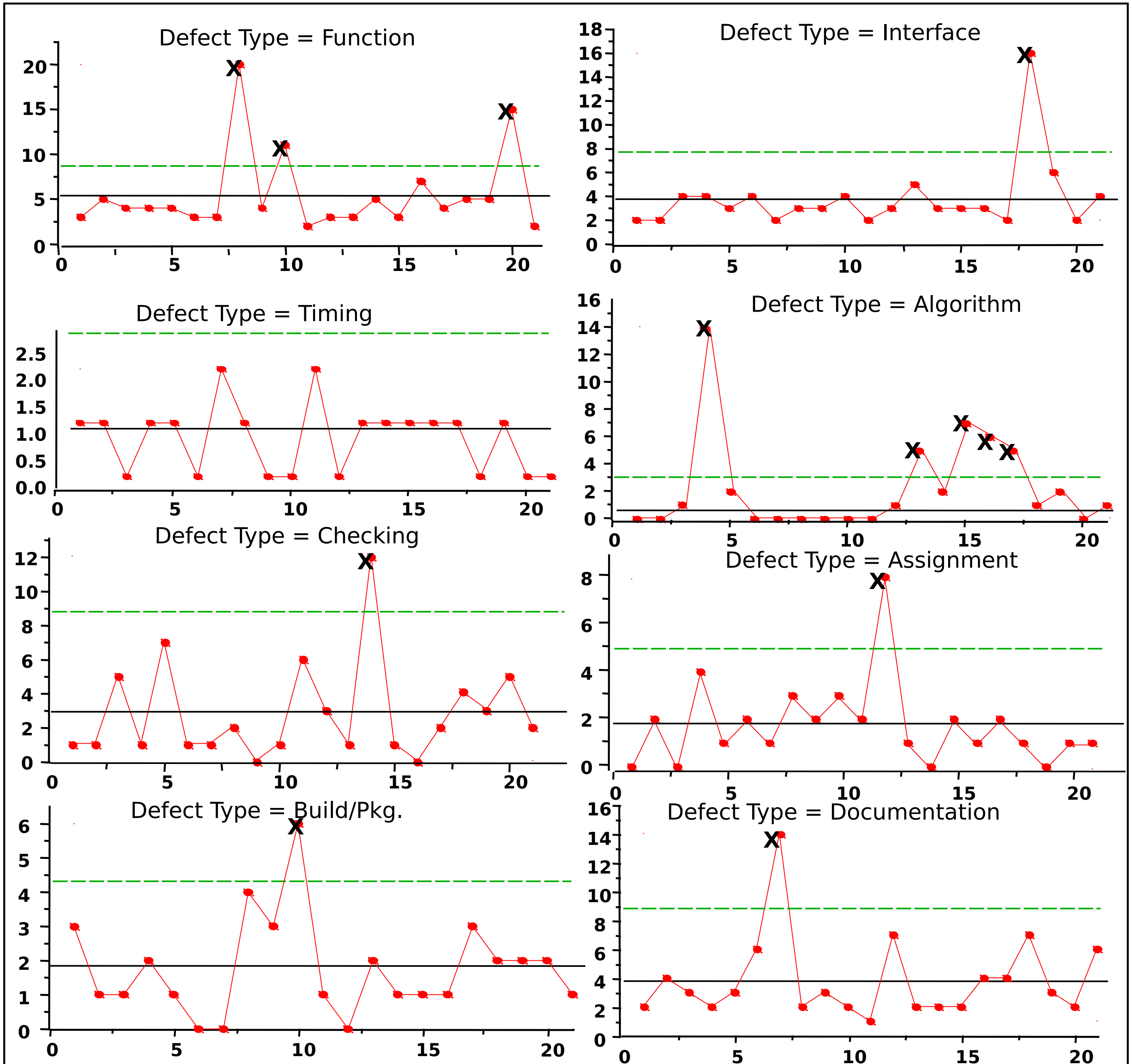
Source: W. Florac and A. Carleton, *Measuring the Software: Statistical Process Control for Software Process Impr*
June 1999.

Control Charts for Individual Defect Types

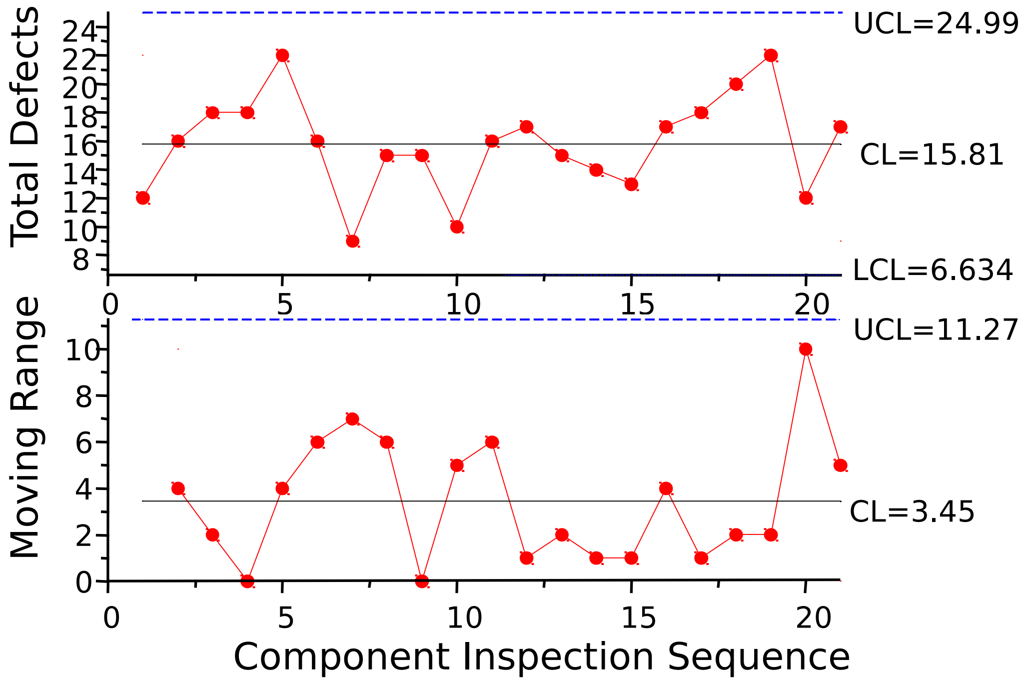


Source: W. Florac and A. Carleton, *Measuring the Software: Statistical Process Control for Software Process Improvement*, Addison-Wesley, Inc., 1999.

Revised Control Charts for Each of the Defect Types

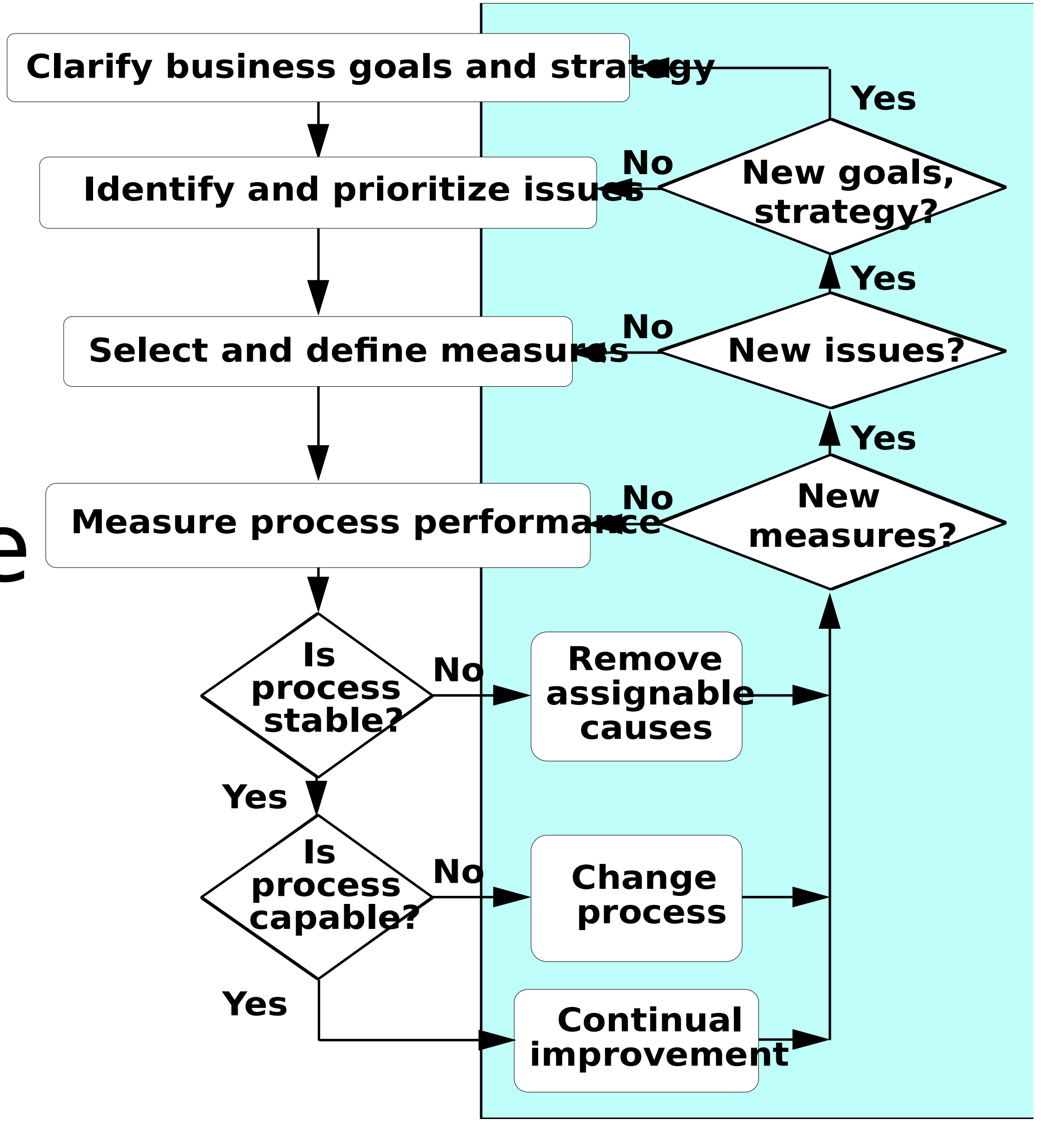


Improved Process-Reduction in Defect Insertion



Source: W. Florac and A. Carleton, *Measuring the Software: Statistical Process Control for Software Process Impr*
June 1999.

Evaluating Process Performance



10 Steps to Get Started - 1

1. Get familiar with SPC techniques; refer to the SEI measurement guidebook and related books and papers on SPC.

2. Obtain a SPC tool.

3. Identify process issues.

4. Identify process performance attributes.

5. Select and define measures.

6. Collect data.

7. Organize the data and ensure principles underlying SPC hold (e.g., homogeniety)

10 Steps to Get Started - 2

8. Plot/graph data.

9. Examine each plot/graph for process stability, process shifts, and assignable causes.

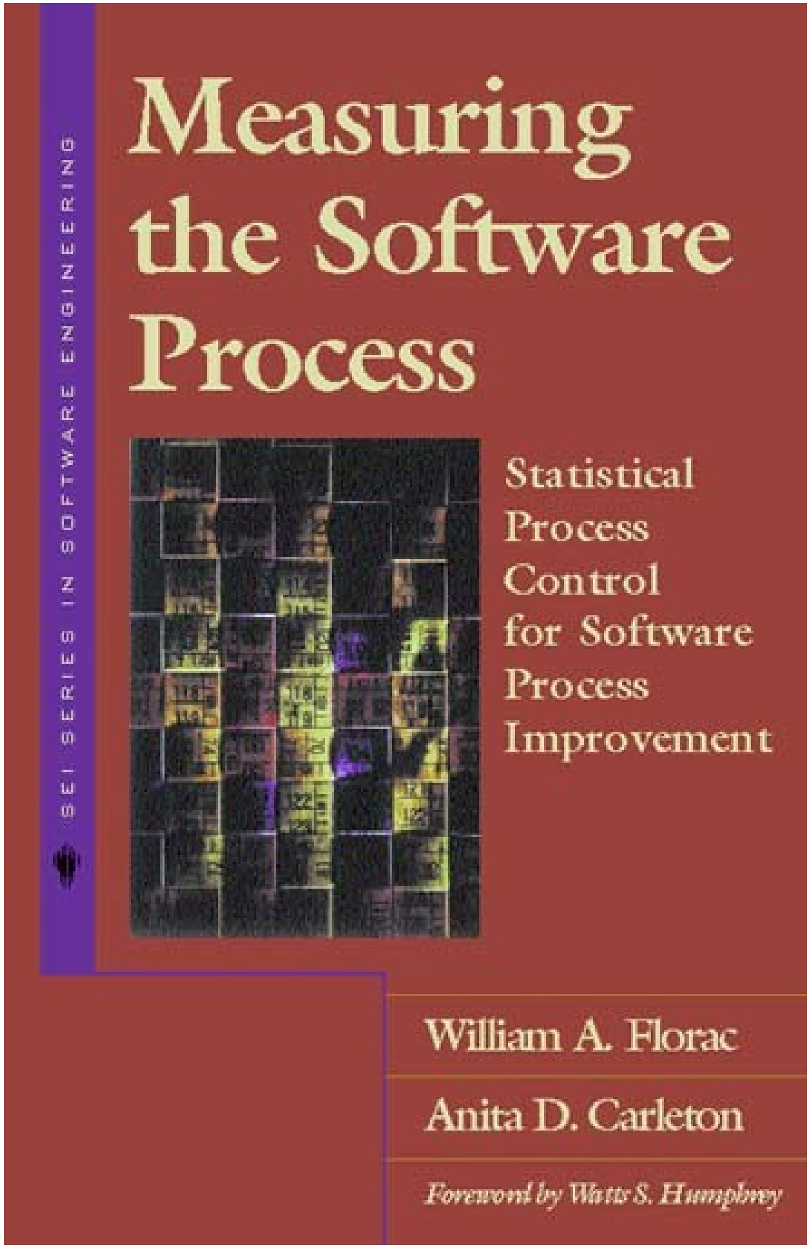
(a) If process NOT STABLE, THEN:

- **cannot determine the capability of the process**
- **no basis to predict outcomes**
- **action: understand why the process is not stable and determine what steps can be taken to achieve stability**

(b) If process is STABLE, THEN:

- **capability of the process can be determined and used to compare future process performance**
- **action: predict future performance and/or examine other process constituents**

10. Run additional analysis as situation requires.


SEI SERIES IN SOFTWARE ENGINEERING
Measuring the Software Process
Statistical Process Control for Software Process Improvement
William A. Florac
Anita D. Carleton
Foreword by Watts S. Humphrey